Observing Cloud Resources

SRE Project Template

Categorize Responsibilities

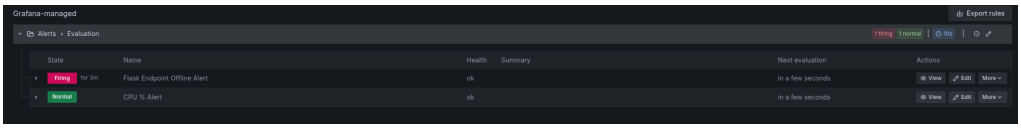
Prometheus and Grafana Screenshots
Provide a screenshot of the Prometheus node_exporter service running on the EC2 instance. Use the following command to show that the system is running: `sudo systemctl status node_exporter`

```
ubuntu@ip-10-100-11-224:~$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-06-23 14:01:51 UTC; 12s ago
 Main PID: 7855 (node_exporter)
    Tasks: 5 (limit: 1109)
   CGroup: /system.slice/node_exporter.service
           └─7855 /usr/local/bin/node_exporter

Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=node_exporter.go:118 level=info collector=time
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=node_exporter.go:118 level=info collector=timex
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=node_exporter.go:118 level=info collector=udp_queues
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=node_exporter.go:118 level=info collector=uname
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=node_exporter.go:118 level=info collector=vmstat
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=node_exporter.go:118 level=info collector=watchdog
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=node_exporter.go:118 level=info collector=xfs
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=node_exporter.go:118 level=info collector=zfs
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=tls_config.go:313 level=info msg="Listening on" address=
Jun 23 14:01:51 ip-10-100-11-224 node_exporter[7855]: ts=2025-06-23T14:01:51.627Z caller=tls_config.go:316 level=info msg="TLS is disabled." http
```

Host Metric (CPU, RAM, Disk, Network)	Dashboard
CPU %	

Memory Dashboard	
Disk I/O	
Network Received	

blackbox	
Alerting rules	

Responsibilities

1. The development team wants to release an emergency hotfix to production. Identify two roles of the SRE team who would be involved in this and why.

Release Manager: This SRE role is directly responsible for overseeing the change management process, managing the deployment pipeline, ensuring all pre-release checks are considered, executing the release, and preparing/executing rollback procedures if needed. Their expertise is crucial for a fast but controlled deployment.
Monitoring Engineer: This SRE role would be critical for immediate post-deployment monitoring. They would closely watch key service metrics, application-specific dashboards, and error logs to quickly identify any negative impact of the hotfix. If the hotfix introduces new issues, they would be the first to respond and potentially initiate a rollback or escalate.

2. The development team is in the early stages of planning to build a new product. Identify two roles of the SRE team that should be invited to the meeting and why.

System Architect: This SRE role should be involved early to understand the proposed architecture, scalability requirements, reliability goals (SLOs), and potential failure modes of the new product. They can provide input on infrastructure choices, data storage, network design, and ensure the design aligns with SRE best practices for resilience and observability from the ground up.
Team Lead: The team lead ensures strategic alignment between the new product's operational demands and the SRE team's overall capabilities, roadmap, and resource availability. They advocate for SRE principles in the design phase, help negotiate achievable SLOs considering supportability, and ensure that the product development lifecycle includes considerations for operational readiness, monitoring integration, and incident response planning from the outset.

3. The emergency hotfix from question 1 was applied and is causing major issues in production. Which SRE role would primarily be involved in mitigating these issues?

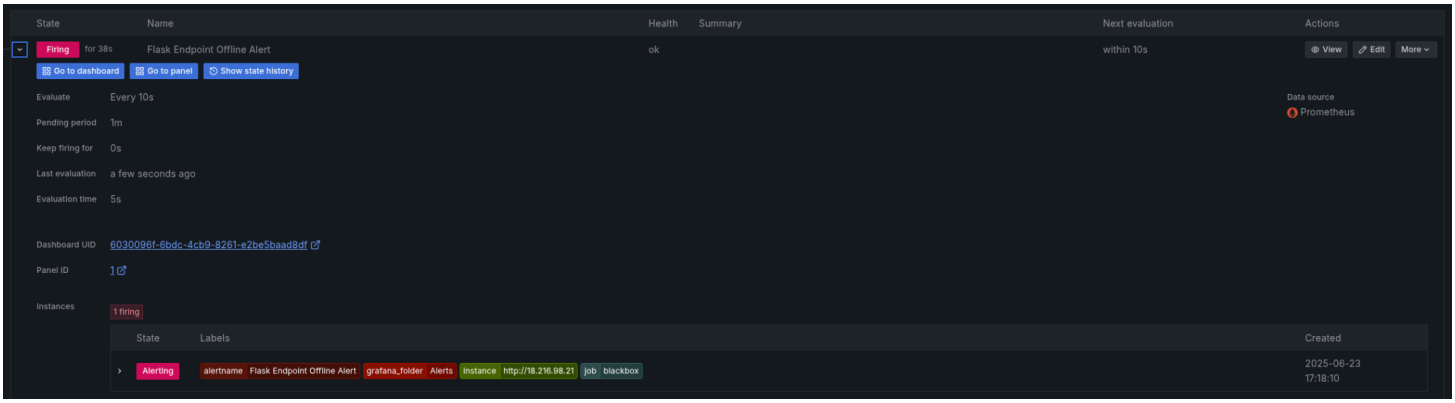
Release Manager: would be heavily involved, especially in executing the rollback of the faulty hotfix.

Team Formation and Workflow Identification

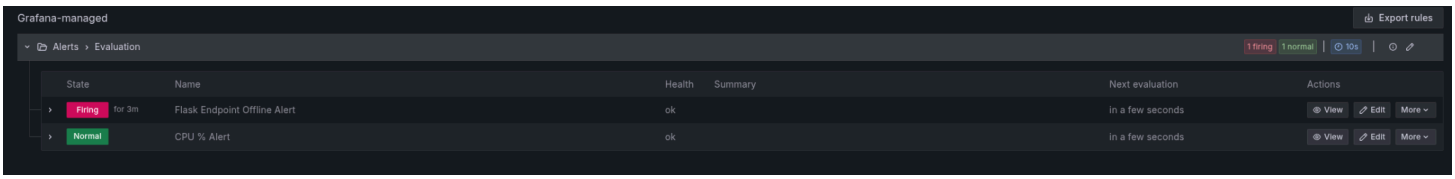
API Monitoring and Notifications

Display the status of an API endpoint: Provide a screenshot of the Grafana dashboard that will show at which point the API is unhealthy (non-200 HTTP code), and when it becomes healthy again (200 HTTP code).

Create a notification channel: Provide a screenshot of the Grafana notification which shows the summary of the issue and when it occurred.

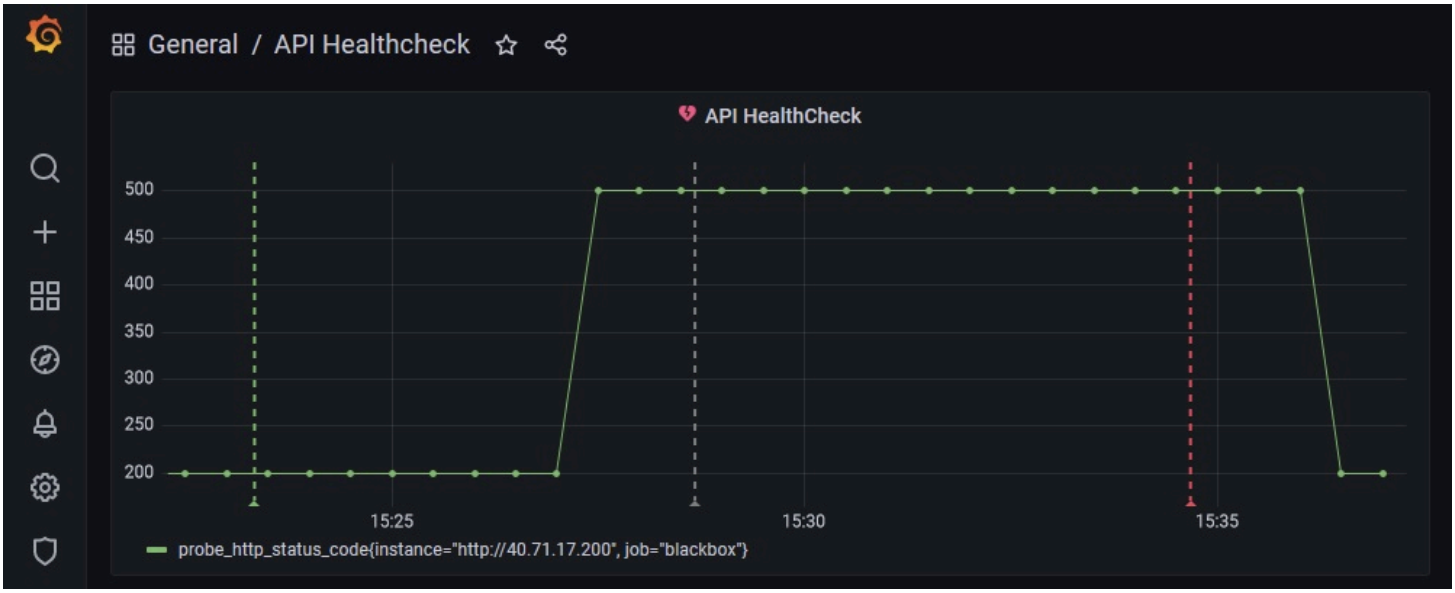


Configure alert rules: Provide a screenshot of the alert rules list in Grafana.



Applying the Concepts

Graph 1



4a. Given the above graph, where does it show that the API endpoint is down? Where on the graph does this show that the API is healthy again?

The API endpoint is shown as **down (unhealthy)** on the graph at 15:28 and fired at ~15:34.
The graph shows the **API is healthy again** from approximately 15:36 onwards.

4b. If there was no SRE team, how would this outage affect customers?

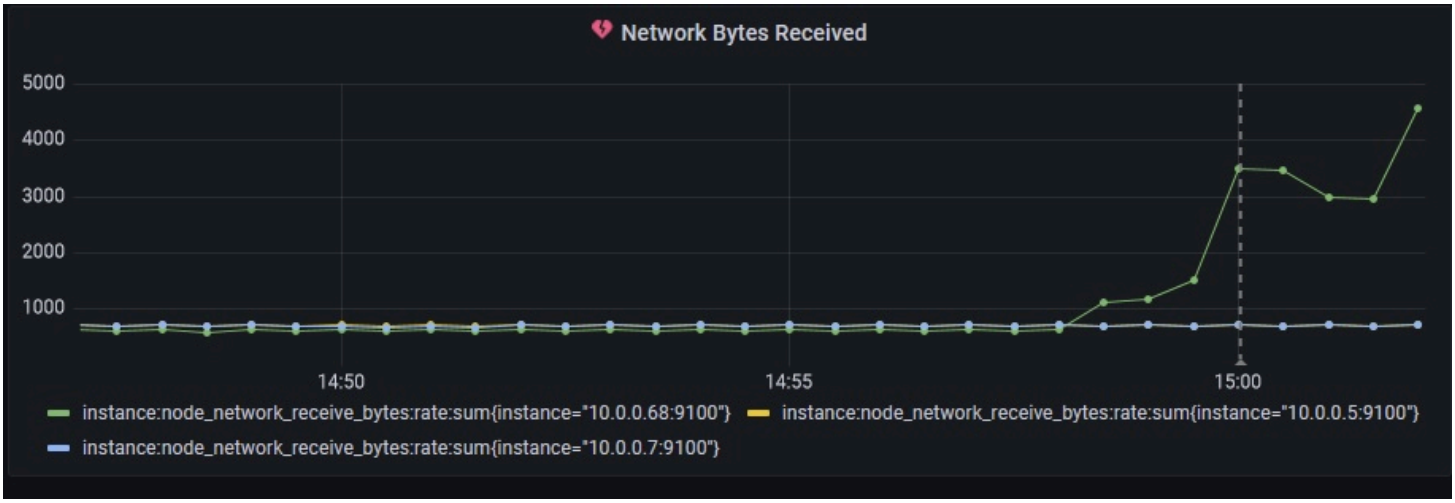
Without an SRE team, the outage would likely go undetected until customers report problems, leading to a significantly longer Mean Time To Detection (MTTD). Consequently, the Mean Time To Resolution (MTTR) would also be extended as ad-hoc troubleshooting by development or operations teams would be slower without established monitoring, alerting, and incident response protocols. This prolonged downtime would directly translate to a degraded customer experience (service unavailability, errors), potential loss of revenue, damage to brand reputation and customer trust, and possible failure to meet contractual Service Level Agreements (SLAs).

4c. What could be put in place so that the SRE team could know of the outage before the customer does?

- Log aggregation and anomaly detection: Centralize application and system logs and implement systems to detect spikes in error messages or unusual patterns that might indicate an impending or ongoing issue.
- Proactive synthetic monitoring: Continuously probe critical API endpoints for availability, correct HTTP status codes (e.g., 200 OK), and expected content or latency.
- Automated alerting: Configure alerts based on these synthetic monitoring metrics (e.g., probe_success == 0, probe_http_status_code != 200, probe_duration_seconds > SLO_threshold). Alerts should trigger rapidly upon confirmed failure and be routed to the on-call SRE.

- Comprehensive application performance monitoring (APM): Instrument the application code to track internal error rates, transaction traces, and resource utilization, with alerts on anomalies or SLO breaches. This can often detect issues before they fully manifest as an external outage.

Graph 2



5a. Given the above graph, which instance had the increase in traffic, and approximately how many bytes did it receive (feel free to round)?

The instance 10.0.0.68:9100 experienced the most significant increase in traffic. It received ~3500 bytes approx.

5b. Which team members on the SRE team would be interested in this graph and why?

Release Manager: They could investigate if the new release introduced code that unexpectedly increased network traffic, which might indicate a bug or an unforeseen side effect requiring a hotfix or rollback.